

REVIEW

Synthetic biology: new engineering rules for an emerging discipline

Ernesto Andrianantoandro^{1,3}, Subhayu Basu^{1,3},
David K Karig^{1,3} and Ron Weiss^{1,2,*}

¹ Department of Electrical Engineering, Princeton University, Princeton, NJ, USA and

² Department of Molecular Biology, Princeton University, Princeton, NJ, USA

* Corresponding author. Department of Electrical Engineering, Princeton University, J-319, E-Quad, Princeton, NJ 08544, USA.

E-mail: rweiss@princeton.edu

³ These authors contributed equally to this work

Received 2.2.06; accepted 17.3.06

Synthetic biologists engineer complex artificial biological systems to investigate natural biological phenomena and for a variety of applications. We outline the basic features of synthetic biology as a new engineering discipline, covering examples from the latest literature and reflecting on the features that make it unique among all other existing engineering fields. We discuss methods for designing and constructing engineered cells with novel functions in a framework of an abstract hierarchy of biological devices, modules, cells, and multicellular systems. The classical engineering strategies of standardization, decoupling, and abstraction will have to be extended to take into account the inherent characteristics of biological devices and modules. To achieve predictability and reliability, strategies for engineering biology must include the notion of cellular context in the functional definition of devices and modules, use rational redesign and directed evolution for system optimization, and focus on accomplishing tasks using cell populations rather than individual cells. The discussion brings to light issues at the heart of designing complex living systems and provides a trajectory for future development.

Molecular Systems Biology 16 May 2006;

doi:10.1038/msb4100073

Subject Categories: synthetic biology

Keywords: context dependence; engineering biology; multicellular system; synthetic biology

Introduction

Synthetic biology will revolutionize how we conceptualize and approach the engineering of biological systems. The vision and applications of this emerging field will influence many other scientific and engineering disciplines, as well as affect various aspects of daily life and society. In this review, we discuss and analyze the recent advances in synthetic biology towards engineering complex living systems through novel assemblies

of biological molecules. The discovery of mathematical logic in gene regulation in the 1960s (e.g. the lac operon; Monod and Jacob, 1961) and early achievements in genetic engineering that took place in the 1970s, such as recombinant DNA technology, paved the way for today's synthetic biology. Synthetic biology extends the spirit of genetic engineering to focus on whole systems of genes and gene products. The focus on systems as opposed to individual genes or pathways is shared by the contemporaneous discipline of systems biology, which analyzes biological organisms in their entirety. Synthetic biologists design and construct complex artificial biological systems using many insights discovered by systems biologists and share their holistic perspective.

The goal of synthetic biology is to extend or modify the behavior of organisms and engineer them to perform new tasks. One useful analogy to conceptualize both the goal and methods of synthetic biology is the computer engineering hierarchy (Figure 1). Within the hierarchy, every constituent part is embedded in a more complex system that provides its context. Design of new behavior occurs with the top of the hierarchy in mind but is implemented bottom-up. At the bottom of the hierarchy are DNA, RNA, proteins, and metabolites (including lipids and carbohydrates, amino acids, and nucleotides), analogous to the physical layer of transistors, capacitors, and resistors in computer engineering. The next layer, the device layer, comprises biochemical reactions that regulate the flow of information and manipulate physical processes, equivalent to engineered logic gates that perform computations in a computer. At the module layer, the synthetic biologist uses a diverse library of biological devices to assemble complex pathways that function like integrated circuits. The connection of these modules to each other and their integration into host cells allows the synthetic biologist to extend or modify the behavior of cells in a programmatic fashion. Although independently operating engineered cells can perform tasks of varying complexity, more sophisticated coordinated tasks are possible with populations of communicating cells, much like the case with computer networks.

It is useful to apply many existing standards for engineering from well-established fields, including software and electrical engineering, mechanical engineering, and civil engineering, to synthetic biology. Methods and criteria such as standardization, abstraction, modularity, predictability, reliability, and uniformity greatly increase the speed and tractability of design. However, care must be taken in directly adopting accepted methods and criteria to the engineering of biology. We must keep in mind what makes synthetic biology different from all previous engineering disciplines. The insight gained from fully appreciating these differences is critical for developing appropriate standards and methods.

Building biological systems entails a unique set of design problems and solutions. Biological devices and modules are

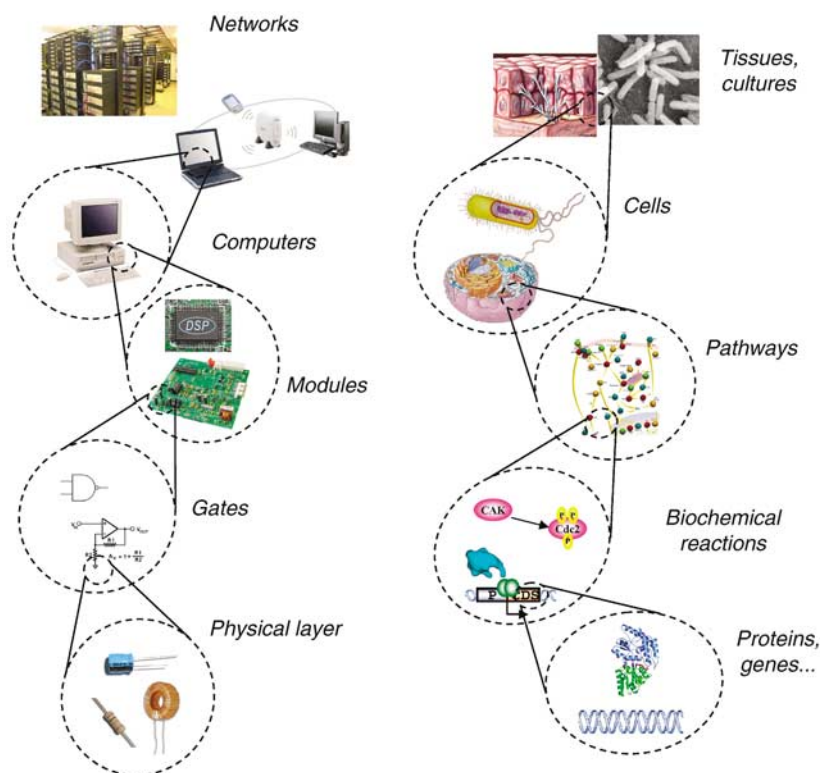


Figure 1 A possible hierarchy for synthetic biology is inspired by computer engineering.

not independent objects, and are not built in the absence of a biological milieu. Biological devices and modules typically function within a cellular environment. When synthetic biologists engineer devices or modules, they do so using the resources and machinery of host cells, but in the process also modify the cells themselves. A major concern in this process is our present inability to fully predict the functions of even simple devices in engineered cells and construct systems that perform complex tasks with precision and reliability. The lack of predictive power stems from several sources of uncertainty, some of which signify the incompleteness of available information about inherent cellular characteristics. The effects of gene expression noise, mutation, cell death, undefined and changing extracellular environments, and interactions with cellular context currently hinder us from engineering single cells with the confidence that we can engineer computers to do specific tasks. However, most applications or tasks we set to our synthetic biological systems are generally completed by a population of cells, not any single cell. In a synthetic system, predictability and reliability may be achieved in two ways: statistically by utilizing large numbers of independent cells or by synchronizing individual cells through intercellular communication to make each cell more predictable and reliable. More importantly, intercellular communication can coordinate tasks across heterogeneous cell populations to elicit highly sophisticated behavior. Thus, it may be best to focus on multicellular systems to achieve overall reliability in performing complex tasks.

Biological devices

Biologists are familiar with manipulation of genes and proteins to probe their properties and understand biological processes. Synthetic biologists must also manipulate the material elements of the cell, but they do so for the purpose of design, to build synthetic biological systems. Synthetic biologists design complex systems by combining basic design units that represent biological functions. The notion of a device is an abstraction overlaid on physical processes that allows for decomposition of systems into basic functional parts. Biological devices process inputs to produce outputs by regulating information flow, performing metabolic and biosynthetic functions, and interfacing with other devices and their environments. Biological devices represent sets of one or more biochemical reactions including transcription, translation, protein phosphorylation, allosteric regulation, ligand/receptor binding, and enzymatic reactions. Some devices may include many diverse reactants and products (e.g. a transcriptional device includes a regulated gene, transcription factors, promoter site, and RNA polymerase), or very few (e.g. a protein phosphorylation device includes a kinase and a substrate). The diverse biochemistries underlying the different devices each provide their own advantages and limitations. Particular device types may be more suitable for specific biological activities and timescales. Although the diversity of biochemical reactions makes it difficult to interface devices, it enables the construction of complex systems with rich functionalities.

We can appropriate unaltered devices from nature, for example, the transcription factor(s) and promoter of a regulated gene, for use in our artificial systems, or we can build devices from modified biochemical reactants. The first synthetic biological devices controlled transcription by modifying promoter sequences to bind novel transcriptional activators and repressors in prokaryotes and eukaryotes (Baron *et al*, 1997; Lutz and Bujard, 1997; Becskei and Serrano, 2000; Lutz *et al*, 2001). Recent efforts also focused on non-transcriptional control. Non-coding RNAs can activate

or silence gene expression by regulating translation events in prokaryotes (Isaacs *et al*, 2004) (Figure 2A). Synthetic biologists also extended the use of non-coding regulatory RNAs to eukaryotes (Bayer and Smolke, 2005). The non-coding RNAs introduced into *Saccharomyces cerevisiae* consisted of aptamer domains that bind specific ligands and antisense domains that target mRNAs. The non-coding RNA undergoes a conformational change upon binding its ligand that enables the antisense domain to bind the target mRNA, thus modulating translation.

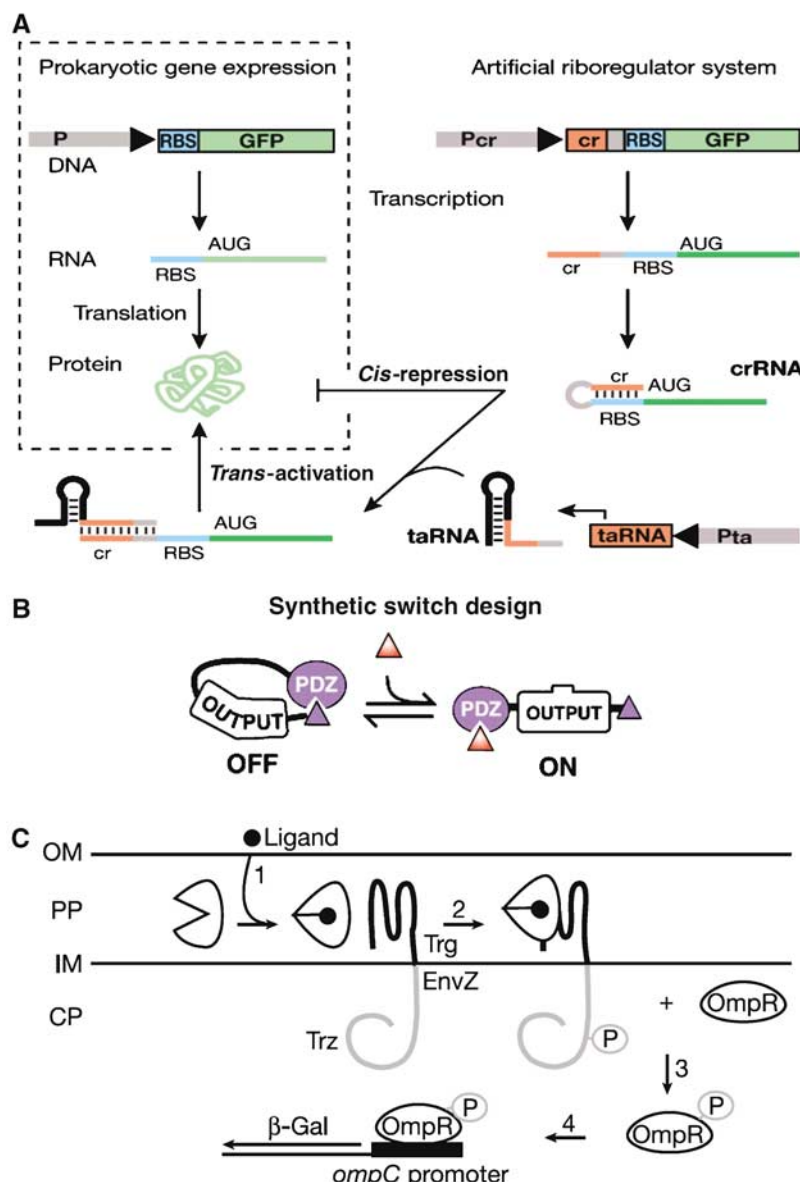


Figure 2 Types of devices. **(A)** Non-coding RNA device. The transcript of a target gene contains an artificial upstream RNA sequence complementary to its ribosome-binding site (RBS), which forms a stem-loop structure in the RBS region, inhibiting translation of the target gene by *cis*-repression. When a transcribed non-coding RNA binds specifically to the artificial *cis*-RNA sequence, this prevents formation of the stem-loop structure in the RBS region, permitting the *trans*-activation of gene expression (Figure 2A reprinted (excerpted) with permission from Isaacs FJ *et al*, *Nat Biotechnol* **22**: 841–847. Copyright 2004 NPG). **(B)** Allosteric protein. The gate is in an 'off' state when an output domain of an engineered protein binds to a tethered allosteric regulatory domain to form an autoinhibited complex. An input ligand can bind to the regulatory domain, relieving the inhibition to liberate the binding or active site of the output domain, switching the gate to the 'on' state (Figure 2B reprinted (excerpted) with permission from, Figure 1B, Dueber JE *et al*, *Science* **301**: 1904. Copyright 2003 AAAS). **(C)** Engineered receptor for trinitrotoluene (TNT), L-lactate, or serotonin. Redesigned *E. coli* periplasmic EnvZ receptors participated in His-to-Asp two-component signaling through autophosphorylation and subsequent transfer of the phosphate to the regulatory response element OmpR (Figure 2C reprinted (excerpted) with permission from Looger LL *et al*, *Nature* **423**: 185–190. Copyright 2003 NPG).

Devices that control transcription and/or translation to produce an output are relatively flexible and easy to build, as nucleotide sequences directly determine the specificity and efficiency of interactions. The ability to concatenate and edit nucleotide sequences at will facilitates the connection of essentially any two devices (as when the promoter in one device is placed in front of the gene encoding a repressor or activator in another device). Transcriptional control systems have many additional useful properties, including signal amplification, combinatorial control by multiple transcription factors, and control of multiple downstream targets. However, changes in output are relatively slow because the process of gene expression occurs on a timescale of minutes owing to the large number of biochemical reactions required to synthesize even a single protein. Achieving a detectable change in output requires many of these protein synthesis events, so these devices consume a large amount of cellular resources during their function.

Devices derived from protein–ligand and protein–protein interactions have different input and output characteristics than transcriptional control devices, and their construction may require more involved modification of their natural substrates. An early approach to building devices based on protein interactions was modular recombination to create allosteric gating mechanisms (Dueber *et al*, 2003) (Figure 2B). Other researchers created protein switches by inserting allosteric domains into existing enzymes (Guntas and Ostermeier, 2004). Random domain insertion followed by directed evolution produced hybrid enzymes (combining maltose-binding protein with beta lactamase) in which maltose altered lactamase catalytic activity by as much as 600-fold (Guntas *et al*, 2005).

Using mutagenesis guided by computational chemistry, protein engineers rationally redesigned and constructed altered periplasmic *Escherichia coli* receptors that respond to a variety of extracellular ligands like TNT, L-lactate, or serotonin instead of their natural ligands (Looger *et al*, 2003), and also converted ribose-binding protein, a sensor protein that lacks enzymatic activity, into a protein that exhibits triose phosphate isomerase activity (Dwyer *et al*, 2004). Construction of these devices utilized computational chemistry to model binding sites and active sites with key residues mutated to confer altered activity (Dwyer and Hellinga, 2004). A geometric search algorithm examines the 3D structure of a protein to locate positions on the polypeptide backbone where placement of prespecified mutated amino-acid side chains simultaneously satisfies all desired geometrical constraints for binding. Designs can be improved iteratively by successive rounds of mutation and geometric search, or by implementing another algorithm that optimizes stereochemical packing of amino-acid side chains. Although these algorithms greatly narrow down the list of requisite mutations, the process must be followed by directed evolution to refine the effectiveness of binding or enzymatic activity. Redesigned receptors can be the initial input devices of extended protein phosphorylation cascades (Looger *et al*, 2003) (Figure 2C).

In constructing protein interaction devices (Giesecke *et al*, 2006), the proteins must be well characterized to determine where changes, deletions, and replacements of domains can

occur, as their 3D structure plays a large role in the nature of their interactions. Connecting protein interaction devices is not a trivial task. Binding interfaces between proteins from different devices must be well matched and one must validate that transfer of information occurs between devices (i.e. a conformational change occurs upon binding or phosphorylation). However, protein interaction devices offer significant design benefits. Binding and enzymatic reactions occur on the sub-second timescale, so changes in output are very fast. It is possible to amplify signals when using protein interaction devices, owing to the reusability of kinases and other enzymes. A minimal amount of protein synthesis can yield devices composed of proteins that undergo repeated interaction events with multiple partners, thus taking up only modest amounts of cellular resources during their function. Additionally, the degree of insulation of protein interaction devices from endogenous cellular processes depends mostly on binding specificity because these devices do not require the protein synthesis machinery of the host cell to produce an output.

Cellular functions of a device are conditioned by the substrate and biochemical reactions chosen. For example, transcriptional and translational devices are easy to connect and are capable of great logical complexity, but such devices cannot be assembled into systems that respond in seconds. Protein interaction devices, however, can provide fast responses. Synthetic biologists will have to combine different types of devices to design the most efficient modules, so future research must establish effective interfacing methods.

Interfacing devices to build synthetic biological modules

A module is a compartmentalized set of devices with interconnected functions that performs complex tasks. In the cell, modules are specific pathways, such as a metabolic pathway or a signal transduction pathway. We must ultimately understand how the function of a module or an entire biological system can be derived from the function of its component parts. Such knowledge will help us establish the biological *rules of composition* to build modules from devices. The rules of composition help determine which device combinations yield the desired logic functions and, more importantly, how to match cellular or physical functions of devices.

Most devices are derived from naturally occurring systems. The difficulty in constructing modules from diverse wild-type devices is that evolution has already optimized them to perform within their natural contexts, so they may not function when connected to each other in an artificial context. Synthetic biologists typically need to change device characteristics (i.e. device physics) in order to produce the desired logical functions when these elements are interfaced. Rational redesign based on mathematical modeling and directed evolution of devices can help match them so they function properly together.

Rational mutations of devices are particularly useful for changing the overall behavior of the system when the properties of these devices are fairly well known. Modifying the kinetics of transcription and translation, operator binding

affinity, and binding cooperativity of transcription factors can help generate devices that enable a module to meet desired criteria, such as having a digital step-like response that yields robustness to extrinsic noise (Baron *et al*, 1997; Weiss and Basu, 2002) (Figure 3). Successful module function thus often requires alteration of wild-type devices to properly interface them.

Even simple modules can take significant amounts of time and resources to construct from devices, often requiring multiple revisions to optimize behavior. Modeling greatly aids in overcoming module design problems (McAdams and Arkin,

1998; Alon, 2003; Kaern *et al*, 2003). The requisite computational tools use abstractions of biochemical reactions to model devices and typically require the rate constants of those reactions. Direct determination of rate constants *in vivo* is still inaccurate and far from trivial, but this barrier may be overcome by the technique of parameter estimation in biological networks (Ronen *et al*, 2002; Braun *et al*, 2005). However, acquiring precise rate constants may not be enough to make predictions if the model is not sufficiently constrained. Adding or removing devices will change the module and change any previously estimated parameters. Thus, parameters derived in one context may not apply in another. We must hence use mechanisms for system redesign and optimization that do not rely on precise parameters.

For system design with incomplete knowledge, it is important to determine which reactions most significantly affect system output. When the system does not function as expected, these reactions are likely to be the best targets for modification. Because the output is typically sensitive to only a few parameters, the set of candidate reactants for mutagenesis can be narrowed to a manageable size. Sensitivity analysis achieves these aims, as demonstrated in a recent study with synthetic gene networks (Feng *et al*, 2004). Sensitivity analysis does not derive actual rate constants, but determines which ones are important. This technique begins with the assignment of random values for rate constants of each reaction and subsequent simulation of the network. The process is repeated with new sets of rate constants until a sufficiently large number of runs are accumulated. Subsequently, one of a variety of algorithms determines the net contribution of rate constants to a previously specified metric of system behavior.

Directed evolution of devices is an alternative and complementary strategy for functionally interfacing devices. If the inherent characteristics of a device are not well known, or if it is unclear how the mutation of particular residues in constituent proteins will affect device behavior, then directed evolution of devices can help convert a non-functional module into a working one. Directed evolution has been used to convert a non-functional two-stage transcriptional cascade (i.e. inverter module) into a working one by mutating the repressor protein and its corresponding ribosome-binding site (Figure 3C) (Yokobayashi *et al*, 2002). Directed evolution resulted in mutations that would have been difficult to design rationally. The major task in implementing directed evolution is deciding what features or behaviors of a system to evolve and applying the appropriate screen or selective pressure. Evolution of complex behavior may not be straightforward. It will take some effort to develop good screens or selection procedures to evolve oscillators, for example. In the future, the combination of rational redesign, parameter estimation, sensitivity analysis, and directed evolution will work together to optimize the functions of connected devices.

Examples of synthetic biological modules

Modules span a continuum of complexity. A simple module may be an instance of a recurring network motif. Complex modules may be composed of many motifs and do not necessarily occur frequently. The variety of motifs allow for

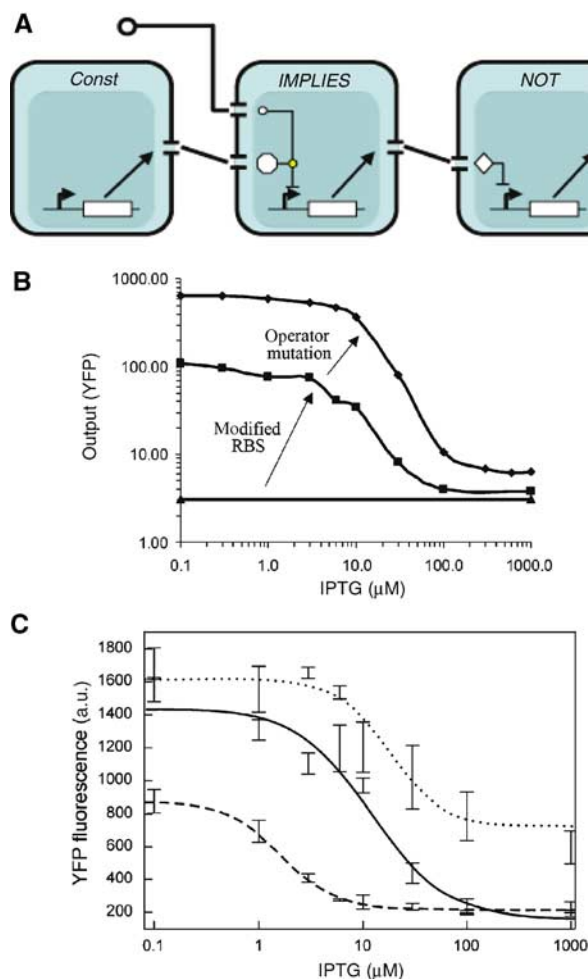


Figure 3 Interfacing devices. **(A)** Transcriptional inverter module with constitutive expression, IMPLIES, and inverter devices. IPTG and LacI are inputs to the IMPLIES device, CI is the input to the inverter device, and YFP is the module output. **(B)** Rational redesign improves inverter module output. The graph shows module output (YFP fluorescence) as a function of input (IPTG concentration). The ideal transfer function of the transcriptional inverter module is an inverse sigmoidal curve. The transfer function is flat and the component is non-responsive when unaltered genetic elements are used in constructing the inverter, but modification of the translational efficiency of the CI protein and further modification of operator binding affinity result in inversely sigmoidal curves with high gain and increased noise margin (Weiss and Basu, 2002). **(C)** Directed evolution offers a complementary redesign strategy for the inverter module. A graph of module output (YFP fluorescence) as a function of input (IPTG concentration) shows that improvement of the transfer function as in panel B can be achieved by directed evolution instead of rational redesign (reprinted from Yokobayashi *et al* (2002); copyright 2002, NAS).

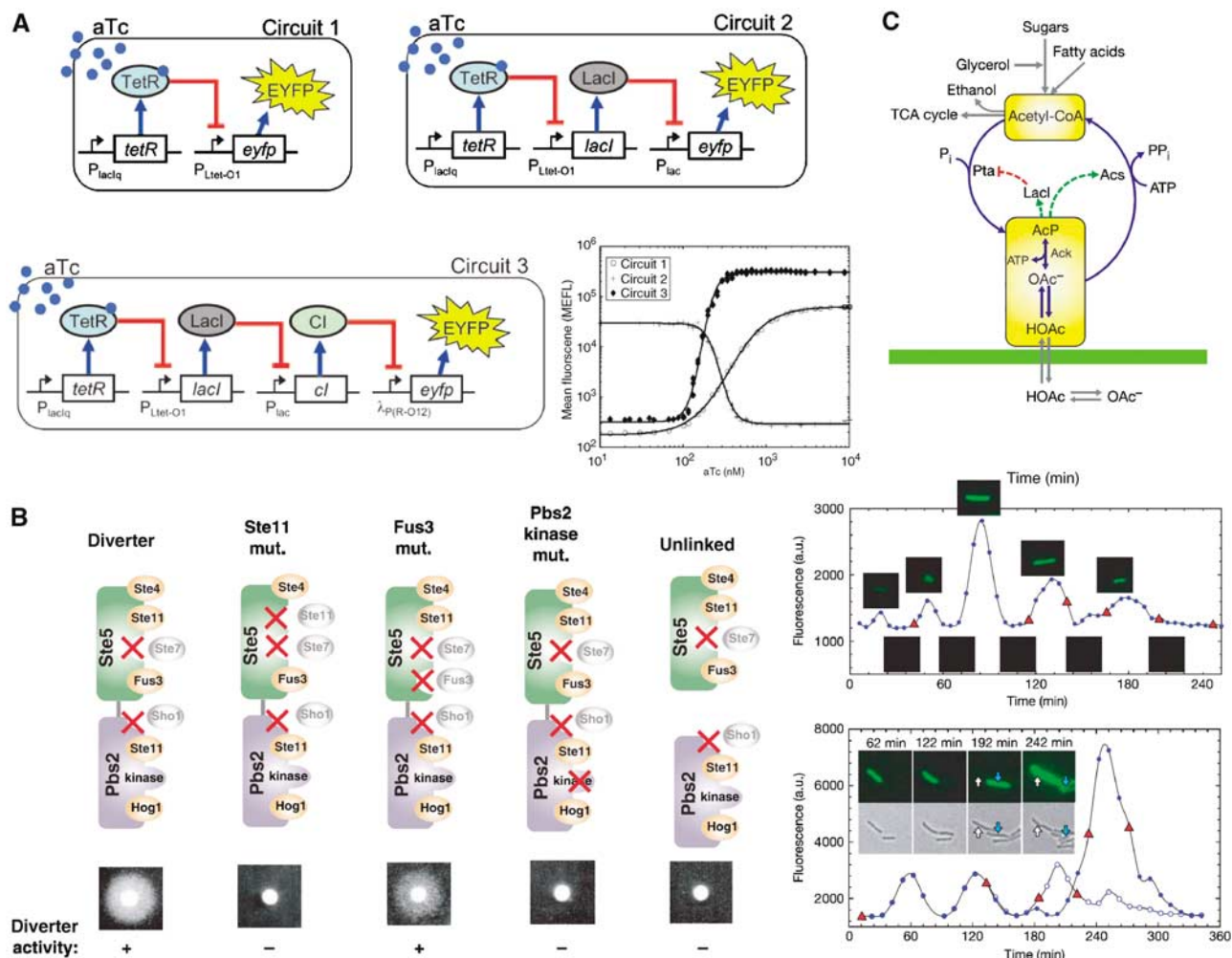


Figure 4 Types of modules. **(A)** Transcriptional cascade modules exhibiting ultrasensitive behavior. Ultrasensitivity increases as a function of cascade depth (reprinted from Hooshangi *et al* (2005); copyright 2005, NAS). **(B)** Schematics of mutant variants of the diverter scaffold. Only the appropriate connectivity (first column) activates HOG1 and allows cell growth (bottom row of micrographs). Growth indicates α -factor-dependent osmoresistance and therefore correct diverter function (Figure 4B reprinted (excerpted) with permission from, Figure 4, Park SH *et al*, *Science* **299**: 1061. Copyright 2003 AAAS). **(C)** The metabolator uses enzymatic reactions that generate acetyl phosphate (AcP) from acetate (HOAc) to elicit oscillations in reporter gene fluorescence under control of the tac promoter (P_{tac}). Graphs depict oscillations in reporter gene fluorescence over time (Figure 4C reprinted (excerpted) with permission from Fung E *et al*, *Nature* **435**: 118–122. Copyright 2005 NPG).

many modes of regulation in natural systems (Milo *et al*, 2002, 2004). It may be tempting to use naturally occurring network motifs and their modules for introducing new behavior into organisms. However, natural modules are not optimized for operation within a cellular context that is not their own, and in fact may not be functional at all. In addition, they are difficult to modify and it may be impossible to find an appropriate natural module that performs a desired task. Synthetic modules can be more readily understood in a quantitative fashion than natural modules. They have well-defined boundaries and points of connection to other modules, and are therefore amenable to insulation from most cellular processes. They can thus be more easily removed, replaced, or altered than natural modules. Most importantly, as we are not limited by natural examples, simple synthetic modules are extensible to functions beyond those available in nature. In this section, we discuss the major prototype synthetic

modules: transcriptional regulation networks, protein signaling pathways, and metabolic networks.

Synthetic transcriptional regulation networks are the most implemented and thoroughly characterized modules to date; they include cascade, feedforward, and feedback motifs. One example of a synthetic transcriptional cascade is a concatenated series of gene transcriptional inverters. Such cascades were constructed in both prokaryotes and eukaryotes (Shen-Orr *et al*, 2002; Blake *et al*, 2003; Rosenfeld and Alon, 2003; Hooshangi *et al*, 2005). The steady-state output of a cascade is monotonic with respect to its input. Under certain conditions, sensitivity of the cascade to a stimulus increases as the cascade depth increases (Hooshangi *et al*, 2005) (Figure 4A). Analysis of the dynamic behavior of one-step and two-step cascades revealed that response delay also has a direct correlation with cascade length (Rosenfeld and Alon, 2003). Response delay can be used for temporal sequencing of gene expression

(Hooshangi *et al*, 2005). Cascades can also attenuate gene expression noise (see Box 1).

Feedforward transcription networks consist of two transcription factors, one of which regulates the other, and both of which regulate a target gene. This configuration allows for a transient non-monotonic response to a step-like stimulus (Mangan and Alon, 2003). Incoherent feedforward modules, where direct and indirect regulation paths respond in opposite ways to input, accelerate response to the step up in stimulus, but not the step down. Coherent feedforward modules, where direct and indirect regulation paths respond in the same way to input, delay response to the step up in stimulus, but not the step down. Appropriate construction can yield modules that act as persistence detectors or delay elements (Mangan *et al*, 2003). Synthetic biologists used incoherent feedforward logic to construct a pulse-generator circuit that displays a transient non-monotonic response to a step up in input stimulus (Basu *et al*, 2004). Modifying the kinetic properties of the constituent devices based on mathematical models yielded pulse generators with different response amplitudes and delays. Pulse generators, persistence detectors, and delay elements can be used to construct circuits with information flowing in parallel at different rates, and with a diversity of behaviors.

Regulatory feedback allows for state and memory functions or internally generated dynamics. Positive feedback can produce bistability, whereas negative feedback can yield

oscillation. A bistable genetic network or 'toggle' switch retains its state even after the removal of input stimulus. A transient chemical or temperature variation flips a genetic toggle switch to express one of its gene products or the other. Pioneering work resulted in a toggle switch constructed using a feedback system composed of two cross-repressing genes (Gardner *et al*, 2000). Researchers used the same feedback mechanism to construct a toggle switch in mammalian cells (Kramer *et al*, 2004). A synthetic oscillatory network constructed by connecting three gene transcriptional repressors, each repressing transcription of another in a ring topology, produced autonomous oscillations (Elowitz and Leibler, 2000). Another synthetic network, based on both positive and negative feedback mechanisms, utilized components of the lactose response system and the nitrogen regulation system of *E. coli* to produce a relaxation oscillator (Atkinson *et al*, 2003).

Synthetic protein signaling pathways have been engineered by modifying and assembling signal transduction devices. Using domain insertion to create synthetic protein devices, one group redirected tyrosine kinase signaling to an apoptotic caspase pathway (Howard *et al*, 2003). The new pathway receives input signals that normally promote cell growth and survival and routes them to output apoptotic responses. Another group (Park *et al*, 2003) created a 'diverter scaffold' that reconfigured the α -factor signal transduction pathway that regulates mating in yeast to instead regulate high osmolarity behavior in yeast (Figure 4B). These developments will lead to the emergence of new synthetic transduction pathways that create novel input-output modules. Protein signaling modules are currently less developed in terms of utilizing varied network motifs than transcriptional regulation modules, but they process cellular and environmental changes quickly and have tremendous potential for use in synthetic systems.

Existing examples of synthetic metabolic networks make use of transcriptional and translational control elements to regulate the expression of enzymes that synthesize and breakdown metabolites. In these systems, metabolite concentration acts as an input for other control elements. An entire metabolic pathway, the mevalonate isoprenoid pathway for synthesizing isopentyl pyrophosphate, from *S. cerevisiae* was successfully transplanted into *E. coli* (Martin *et al*, 2003). In combination with an inserted synthetic amorpha-4, 11-diene synthase, this pathway produced large amounts of a precursor to the anti-malarial drug artemisinin. In addition to engineering pathways that produce synthetic metabolites, artificial circuits can be engineered using metabolic pathways connected to regulatory proteins and transcriptional control elements. One study describes such a circuit based on controlling gene expression through acetate metabolism for cell-cell communication (Bulter *et al*, 2004). Metabolic networks may embody more complex motifs, such as an oscillatory network. A recently constructed metabolic network used glycolytic flux to generate oscillations through the signaling metabolite acetyl phosphate (Fung *et al*, 2005). The system integrates transcriptional regulation with metabolism to produce oscillations that are not correlated with the cell division cycle (Figure 4C). The general concerns of constructing transcriptional and protein interaction-based modules, such as kinetic matching and optimization of reactions for

Box 1 | Noise

Biological systems are inherently noisy, and must either minimize or take advantage of stochastic fluctuations to maintain function. Pioneering work on the oscillator showed that gene expression noise from variations in intrinsic and extrinsic factors can prevent synthetic modules from working cohesively *in vivo* to execute desired functions (Elowitz *et al*, 2002). Intrinsic factors governing noise in gene expression include stochasticity in biochemical reactions at the individual gene level, whereas extrinsic factors include fluctuations in cellular components such as metabolites, ribosomes, and polymerases. Researchers have used synthetic transcriptional cascades to study noise propagation in genetic networks and examine the effect of intrinsic and extrinsic factors on phenotypic variations (Thattai and van Oudenaarden, 2001; Elowitz *et al*, 2002; Blake *et al*, 2003; Hooshangi *et al*, 2005; Rosenfeld *et al*, 2005). Given the ubiquity of noise in biological systems, what noise levels are acceptable for reliable operation? The amount of allowable noise will depend on the timescale and nature of the operation. In cells, fast and slow fluctuations affect the operation of modules in different ways. Rosenfeld *et al* (2005) found that the fluctuations due to intrinsic noise of gene expression were much faster than the cell cycle, whereas fluctuations due to extrinsic noise occurred on a longer timescale and had significant effects on function. Careful design can attenuate noise, as some network architectures are more robust to noise than others. For example, transcriptional cascades are more robust to extrinsic noise than a single gene transcriptional inverter because small fluctuations in the input are not transmitted to the output (Hooshangi *et al*, 2005). The related phenomenon of ultrasensitivity confers an approximate digital response to networks. Digital components allow for the construction of large, complex systems that remain digital and have predictable, robust behavior. Noisy analog components are difficult to assemble into complex systems because noise may be amplified in analog systems. A system can also manage noise by incorporating noise into its functions. Population control is an example of a synthetic system that uses noise for correct function (You *et al*, 2004), whereas natural examples are lambda (Ptashne, 1986) and HIV (Weinberger *et al*, 2005). Accounting for noise during modeling and simulation of device and module function will be important for designing biological systems that perform as expected. Conversely, synthetic systems will also provide good models to study noise in biology.

a new environment, apply for metabolic networks as well. In addition, the appropriate metabolic precursors must be present. For this purpose, it may be necessary to include other enzymes or metabolic pathways that synthesize precursors for the metabolite required in a synthetic network.

Tremendous progress has been made in developing prototype modules with non-trivial behaviors. Modules with certain motifs have been shown to function as desired (e.g. ultrasensitive cascades, feedforward motifs, and bistable switches). These motifs have been engineered to operate correctly using the same tools applied to connecting devices, as described in the previous section. Much work is still needed to engineer certain other motifs to function reliably and predictably, for example, oscillators. Future research will focus on the integration of these basic motifs to form complex modules with interesting high-level functions (counters, adders, multiple signal integrators). Proper interfacing of diverse module types (transcription regulation, protein signaling, and metabolic networks) can extend their function and such procedures will make reliable, definable connections to cellular context and thus generate well-designed cells.

Engineering cells: interfacing with cellular context and connecting modules

The functional behavior of a module in a cell depends not only on its component devices and their connectivity (wiring), but also on the cellular context in which the module operates (Figure 5A). Relevant cellular context includes general biochemical processes such as DNA and RNA metabolism, availability of amino acids, ATP levels, protein synthesis, cell cycle and division, and specific processes such as endogenous signaling pathways that may interact with devices in the exogenous module. As a consequence, the same gene circuit can have different behavior in slightly different host cell strains (Endy, 2005). In addition, integration and function of a module in a host cell may fundamentally affect host cell processes, thus altering the cellular context, which may then recursively alter the behavior of the module. The situation is further complicated by the integration and function of additional modules.

Because synthetic modules and endogenous cellular processes condition each other's behavior, any fluctuations in the host cell processes are relayed to the module and affect its output and vice versa. This presents a problem for engineering predictable, reliable biological systems. One approach to solving this problem is to take the notion of modularity to heart. Modularity is used in other engineering disciplines to insulate interacting systems from each other and render them interchangeable. According to this notion, inserted modules would function best if the number of interactions between the module and the host cell are minimized. Any remaining interactions should ideally be very predictable. Specifying and standardizing those remaining interactions can ensure the portability of the modules, and allow them to be engineered independently of host cells. Using this approach, module function would ideally become independent of cellular context. The host cell would only act to process resources and protect the module from the extracellular environment.

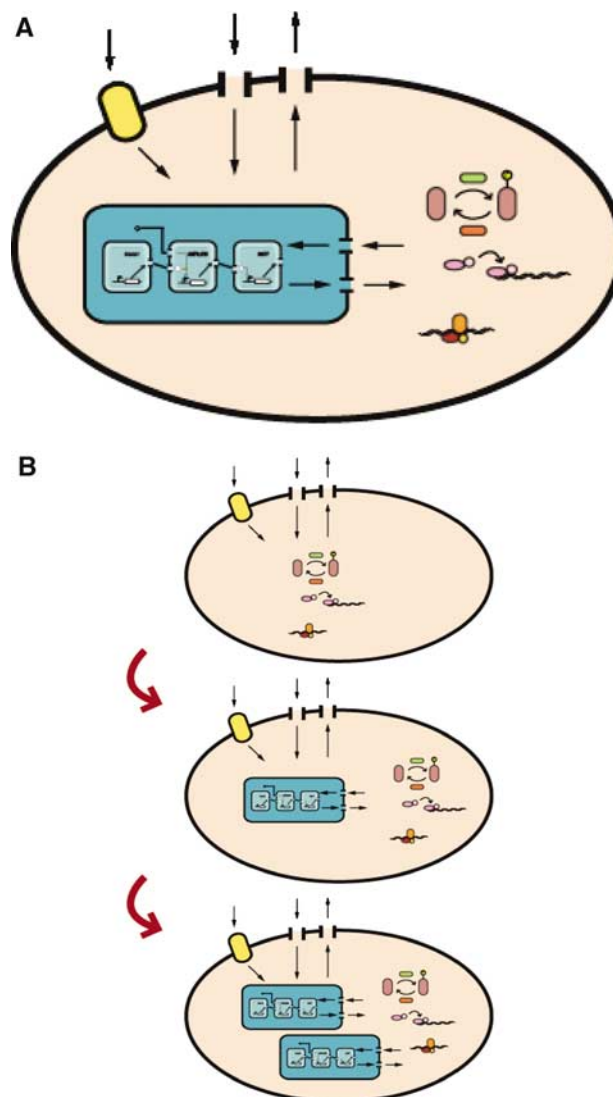


Figure 5 Context dependence. (A) Modules operate within and modify the cellular context. (B) Successive insertions of modules recursively modify cellular context such that each new module is embedded in a new context, perhaps fundamentally altering module behavior.

If achievable, insulation of modules is useful, but must be tempered with a drive to understand and take into account a module's connection to the host's cellular context. Although simplification, specification, and standardization make engineering easier, it may not be advantageous to hide all the information about the host cell. Conceptualizing the operation of a module as completely, or nearly completely, disconnected from cellular context cannot sufficiently define module function. Part of what defines living systems is the integration of their parts. Engineering any part of an organism must at some level take the entire organism into account. Thus, for modular composition to work, we need to have abstractions that incorporate the notion of cellular context into the definition of a module's function. We must have a quantifiable way to encode context dependence for a given module and functionally compose the context dependence of multiple modules.

Rational redesign, directed evolution, and modeling, which played such important roles in assembling devices into modules, can play similar roles in interfacing modules with the cellular context of a host cell. Combining parameter estimation techniques with metabolic flux balance analysis to take into account relevant contextual features may be a promising path. It may be necessary to quantify the effects of context iteratively after the addition of each module to glean a more accurate description of cells that harbor our modules of interest (Figure 5B). Insertion of some modules may be easy, but others will be difficult to insert and will require a great deal of modification of the host cell for optimal compatibility. Adding multiple modules piecemeal in this fashion could be prohibitively difficult. The parallel with software engineering is instructive: this is like adding foreign software to an operating system (OS), each new program with its own patch. Adding too many programs often leads to system-wide instability. A better strategy in this case would be to build a new OS. For synthetic biology, this means engineering a new organism by synthesizing its genome.

Synthetic genomes

Advances in DNA synthesis (see Box 2) will drive progress in the construction of synthetic genomes to provide a reliable method for building an entirely artificial organism (Zimmer, 2003). Constructing an organism wholesale has certain advantages. We may choose to include only the functions and pathways that we want, either for simplicity, to cut down on evolutionary baggage, or to make the genome easier to customize. The system would allow users to plug in any desired module, and implement hooks for extensibility of features and perhaps dependence on laboratory supplements in the media for safety. A synthetic genome, like a computer's

Box 2 | DNA synthesis

The ability to synthesize large DNA sequences in a fast, inexpensive, and accurate manner will have a great impact on the size and complexity of synthetic gene networks and will enable the synthesis of whole genomes. Methods utilizing PCR to assemble pools of short-length oligos have been successfully used to construct gene sequences of over 14 kb (Stemmer *et al*, 1995). For example, a contiguous 32 kb gene cluster containing the *E. coli* polyketide synthase genes was constructed by creating 500 bp segments through PCR of a short oligo pool, connecting these 500 bp segments to 5 kbp segments through 'ligation by selection' (Kodumal and Santi, 2004), and finally using conventional cloning methods to assemble the 5 kbp segments into the final 32 kb gene cluster (Kodumal *et al*, 2004). However, mutations can arise in oligonucleotide synthesis as well as in the PCR steps. Thus, error correction procedures are necessary for constructing large, high-fidelity sequences. A recent approach uses photo-programmable microfluidic chips to produce one library of oligonucleotides for constructing large DNA sequences and another library of oligonucleotides for error correction (Tian *et al*, 2004). The error correction oligos hybridize to the correct construction oligos, whereas most incorrect oligos do not hybridize and are washed away. This procedure has been used to successfully synthesize the 21 *E. coli* 30S ribosomal subunit genes. Other error correction procedures include the use of proteins such as MutS to detect DNA mismatches and separate correct and incorrect sequences (Carr *et al*, 2004). Combining these approaches will enable the efficient and inexpensive production of high-fidelity megabase-scale sequences. The capacity for large-scale DNA synthesis will allow for decoupling of design from fabrication in engineering genetic modules.

OS, needs particular conditions and substrates to operate, but may incorporate only desired features. Just as with operating systems, synthetic genomes would be delivered with manuals and full documentation. If there is no simple migration path from one genome or OS to another, then it would be simpler to change the OS, or synthesize a novel genome than to try and convert multiple applications or modules. A good place to start is with the simplest known genomes, that is, viruses. Synthetic virus genomes have been constructed *de novo* and successfully tested for the ability to generate infectious viruses. Poliovirus cDNA was synthesized, transcribed into viral RNA, replicated in cell extracts, and injected into transgenic mice (Cello *et al*, 2002). The synthetic poliovirus induced the same neurovirulent phenotype as wild-type poliovirus. The ϕ X174 bacteriophage genome (5386 bp) has also been synthesized, and used to infect *E. coli* cells (Smith *et al*, 2003). In other work, the T7 bacteriophage genome has been refactored to make the virus simpler to model and more amenable to manipulation (Chan *et al*, 2005).

Beyond viruses, progress has been made towards identifying and removing non-essential genes in the genomes of *E. coli* as well as *Mycoplasma genitalium*, which has one of the smallest known genomes (Pennisi, 2005). The resulting minimal genomes may yield organisms that serve as ideal vessels for synthetic gene networks. Genome synthesis will make it possible to fabricate minimal cells, creating the simplest possible contexts for inserting new modules. This leads to the question: can we design a better 'cell chassis'? We still need to determine if the properties of a minimal cell chassis are those of an optimal cell chassis. Simplification of its genome may not yield a generic organism that operates with precision and reliability. Also, the problem of context dependence is not eliminated by the construction of a generic cell chassis, because the addition of any module changes the nature of that chassis. Nonetheless, the chassis simplification will make it easier to quantify and model cellular context. Although it is a useful goal to make chassis cells uniform, this still does not guarantee operational reproducibility. Uniformity will be difficult to maintain in the face of mutation and evolution of the cell lines. Although the lure to construct minimal genomes is strong, it may be more advantageous to implement less compact genomes. Continuing the software analogy, a faster, compact OS may only have a few applications and software libraries, whereas a slower, bulkier OS may have many libraries and applications. Such bulky systems may in fact offer more reliability through additional, perhaps redundant mechanisms. Whichever OS you are using, software cannot run without the computer. We endorse a perspective in which the objects of design are not just modules and devices, but modules and devices embedded in host cells. The complex biological system engineered to perform a task should be the cell, and in fact not just one cell, but a *population* of cells.

Composition of multicellular systems

Synthetic single cells may become easier to generate in the future, but there are still physical limitations to the type of complex tasks even large numbers of completely independent cells can accomplish and how reliably they perform those

tasks. Distributing synthetic networks among multiple cells to form artificial cell–cell communication systems both increases the number of design possibilities and can overcome the limited reliability of individual synthetic cells. Even populations of genetically identical cells exhibit heterogeneous phenotypes and constituent cells behave asynchronously owing to intrinsic and extrinsic noise in gene expression and other cell functions. A group of non-communicating cells will not behave identically, let alone in a coordinated fashion. Utilizing cell–cell communication to coordinate behavior can overcome the problem of asynchronous behavior of engineered cells in populations as well as multicellular systems. One of the first efforts to engineer multicellular systems in synthetic biology by assembling and recombining parts was the development of cell–cell communication modules in *E. coli* to coordinate the behavior of cell communities (Weiss, 2000). The genes responsible for quorum sensing in *Vibrio fischeri* (Miller and Bassler, 2001) were compartmentalized into separate sender and receiver cell types. Sender cells expressed LuxI, which catalyzes the synthesis of small, diffusible acyl-homoserine lactone (AHL) molecules. These AHL signal molecules diffuse to the receiver cells, in which they form a complex with the LuxR protein and activate reporter transcription. Researchers have since increased the sensitivity of *V. fischeri* LuxR for a variety of derivatives of AHL and identified new LuxR residues important for ligand binding (Collins *et al*, 2005).

More recent work saw advances in utilizing increasingly complex network architectures. By interfacing quorum-sensing behavior with programmed cell death in *E. coli* cells, the population can be kept at a tunable low density for long durations despite (and in fact owing to) noise in the system (You *et al*, 2004; Balagadde *et al*, 2005) (Figure 6A). If cells in the population control system behaved in a fully synchronous digital fashion, simultaneous death of all cells would occur once the threshold quorum-sensing level was reached. Instead, variations in response of individual cells due to noise allow some cells to survive, keeping the population at a tunable low density. The use of intercellular communication to coordinate behavior was also suggested to be effective in synchronizing oscillations and improving robustness to gene expression noise (McMillen *et al*, 2002; Garcia-Ojalvo *et al*, 2004). Attempts to construct such complex synthetic multicellular systems will shed light on the key features of natural networks that contribute to robust behavior and will reveal how these networks tolerate or even take advantage of noise.

Moving from single cells to multicellular systems opens new vistas for exploration: the spatial aspects of the behavior of biological systems. Synthetic biologists compartmentalized portions of a pulse-generating network into separate sender and receiver cell populations where sender cells produce the signal that triggers a pulse in the receiver cells. Sender cells were placed in a fixed position with receiver cells surrounding them on solid-phase media. As the pulse-generating network can sense different rates of increase of the signal, receiver cells near the sender cells produce a detectable pulse response, whereas cells farther away did not respond as well. This signal processing capability is useful for designing systems that require communication mechanisms with fine-tuned localized responses to a diffusible signal. Researchers also demonstrated

programmed pattern formation using a band-detect network (Basu *et al*, 2005). The network integrates an artificial cell–cell communication system with multiple transcriptional genetic inverters that have dissimilar repression efficiency in a feedforward motif. The system only responds to a predefined range of signal concentration, and this range is engineered by altering different parameters of the network. By placing sender cells in predetermined positions with variants of undifferentiated band-detect cells surrounding them on solid-phase media, the band-detect system generated elaborate spatial patterns in the form of hearts, clovers, bullseyes, and ellipses (Figure 6B). More complex multicellular patterns with enhanced robustness will be achieved by integrating multiple communication signals and feedback mechanisms into the system.

We face important challenges in constructing more sophisticated multicellular systems. Communication between many different types or populations of cells requires not only insulation between multiple communication channels, but also interoperability. Building such complexity will require the use of several types of signaling components, more kinds of host cells, multiway communication, and improvements in precision and reliability. Designing communication systems in a multicellular environment entails balancing the sensitivities of the intercellular elements and reducing the crosstalk between those signals. Researchers recently constructed a signal amplification network to improve the sensitivity of communication systems from *Pseudomonas aeruginosa* and analyze crosstalk among different quorum-sensing signals (Karig and Weiss, 2005). One effort to develop intercellular communication elements distinct from existing AHL-based quorum sensing turned to the metabolite acetate as a central element (Bulter *et al*, 2004). This system differs appreciably from existing AHL-based systems and should produce minimal crosstalk, making it useful for designing robust multicellular systems. Recently, multicellular communication modules were also developed in yeast to achieve sender–receiver and quorum-sensing behaviors (Chen and Weiss, 2005). The systems integrated *Arabidopsis thaliana* signal synthesis and receptor components with endogenous yeast protein phosphorylation elements and new response regulators (Figure 6C). Engineered yeast cells synthesize the plant hormone cytokinin in a positive feedback loop to achieve quorum sensing. Cytokinin diffuses into the environment and activates a hybrid exogenous/endogenous phosphorylation network that triggers production of more cytokinin, leading to population-dependent gene expression. These initial strides will build the foundations for inter-organism communication, which will allow for a greater diversity of future applications.

The design, construction, and testing of multicellular systems will ultimately have a significant impact on central problems both in biology and computing. Synthetic biology provides unique opportunities and powerful tools to investigate spatiotemporal patterning of gene expression and the mechanisms governing coordinated cell behavior during development of multicellular organisms. We will eventually be able to build synthetic model systems of biological development. Synthetic biology will provide useful and manipulatable model systems to answer a central question in computing and biology: how can complex and robust global

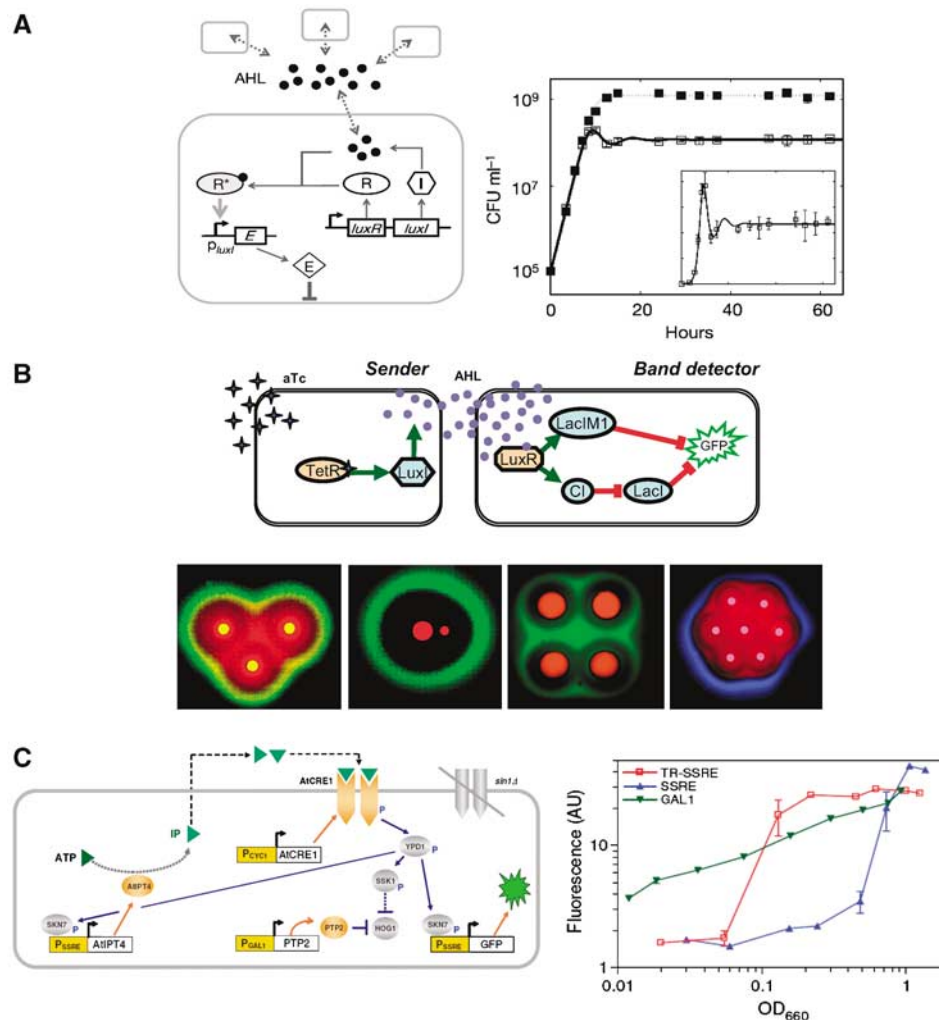


Figure 6 Multicellular systems. **(A)** A population control circuit programs an AHL synthesizing cell culture to maintain an artificially low cell density through AHL-induced cell death. The graph shows colony forming units in the cell culture as a function of time for circuit 'ON' (open squares) and circuit 'OFF' (filled squares). The inset shows damped oscillations that occur before stabilization of culture density (Figure 6A reprinted (excerpted) with permission from You L *et al*, *Nature* **428**: 868–871. Copyright 2004 NPG). **(B)** Programmed pattern formation system with sender cells and band detect cells that respond to prespecified concentrations of AHL. The pictures depict a variety of patterns formed by the placement of sender disks in various locations (Figure 6B reprinted (excerpted) with permission from Basu S *et al*, *Nature* **434**: 1130–1134. Copyright 2005 NPG). **(C)** Artificial cell–cell communication in *S. cerevisiae* using communication elements from *A. thaliana*. The positive feedback motif results in quorum-sensing behavior that can be fine-tuned based on the regulatory mode for signal synthesis. The graph depicts output (GFP fluorescence) as a function of cell population (optical density) where the signal (IP) synthesis rate is controlled by expression of *AtIPT4* enzyme under different promoters: unregulated (green), weaker basal expression with positive feedback (red line), and stronger basal expression with positive feedback (red line) (Figure 6C reprinted (excerpted) with permission from, Figure 1A, Chen MT and Weiss R, *Nat Biotechnol* **23**: 1552. Copyright 2005 NPG).

behavior emerge from the interactions of large numbers of unreliable, locally communicating components?

Applications

Bacteria and yeast are already used in numerous biotechnology applications such as fermentation and drug synthesis. The ability to engineer multicellular systems will drive vast improvements in existing applications as well as open up a wide variety of new possibilities. For example, programmed coordination of engineered bacteria or yeast can eliminate the need for monitoring of batch cultures and control of gene expression in such cultures through addition of expensive

inducers (Farmer and Liao, 2000; Chen and Weiss, 2005). New applications can be obtained by coupling gene regulatory networks with biosensor modules and biological response systems. Researchers interfaced a toggle switch with the SOS pathway detecting DNA damage as the biosensor module and biofilm formation as response output (Kobayashi *et al*, 2004). Exposure of engineered cells to transient UV irradiation caused DNA damage, triggering the toggle switch to flip to the state that induced the formation of biofilms. In a similar study, a toggle switch was interfaced with the transgenic *V. fischeri* quorum-sensing module to sense when cells reach a critical threshold density. Engineering interactions between programmed bacteria and mammalian cells will lead to exciting medical applications. Recently, *E. coli* cells were engineered to

invade specific mammalian cells exhibiting tumorigenic properties. Different plasmids were built containing the *inv* gene from *Yersinia pseudotuberculosis* under control of the Lux promoter, the hypoxia-responsive *fdhF* promoter, and the arabinose-inducible *araBAD* promoter. *E. coli* harboring these plasmids invaded cancer-derived cells in a density-dependent fashion, under anaerobic growth conditions, and upon arabinose induction, respectively (Anderson *et al*, 2005). Synthetic biology will help the treatment of disease move beyond traditional approaches by allowing us to develop 'smart' therapies, where the therapeutic agent can perform computation and logic operations and make complex decisions. Also, the ability to engineer synthetic systems that can form spatial patterns is a critical step towards tissue engineering and fabrication of biomaterials (Basu *et al*, 2005). Synthetic biology will open the doors to promising new applications, such as the production of alternative energy sources in live cells, biological computing, and bio-directed fabrication.

Conclusions

Synthetic biology distinguishes itself from other engineering and scientific disciplines in both its approach and its choice of object. This emerging field uses the insights of scientific biological inquiry but formulates new rules for engineering purposes. Synthetic biology should be considered a hybrid discipline, combining elements of both engineering and science to achieve its goal of engineering synthetic organisms.

Living systems are highly complex, and we currently lack a great deal of information about how these systems work. One reason is that biological systems possess a degree of integration of their parts far greater than that of non-living systems. Breaking down organisms into a hierarchy of composable parts, although useful as a tool for conceptualization, should not lull the reader into thinking that these parts can be assembled *ex nihilo*. Because we do not yet know how to confer the properties of life onto an aggregate of physically dynamic, but 'dead' material systems, composing artificial living systems requires the use and modification of natural ones. Therefore, assembly of parts occurs in a biological milieu, within an existing cellular context. This has profound implications for the abstraction of biological components into devices and modules and their use in design.

In computer engineering, it is possible to isolate hardware design (computer architecture) from software design (programming), making it easy to implement different behaviors on the same physical platform. The software analogy used earlier was instructive for understanding the development of synthetic genomes and the role of modules within that framework. However, at this stage of synthetic biology, 'programming' actually means altering the hardware itself. Reprogramming a cell involves the creation of synthetic biological components by adding, removing, or changing genes and proteins. Direct interaction with the hardware of engineered cells thus more closely resembles the incorporation of a user-designed application-specific integrated circuit (ASIC) into a computer. With ASICs, users can easily extend computer function by designing specialized hardware logic that is then fabricated on a custom silicon chip. In contrast to

writing computer software, extending computer hardware by incorporating ASICs may require modifications to existing hardware (e.g. adding an extra power supply or enlarging the computer chassis). Similarly, the addition of synthetic gene networks (new cellular hardware) to host cells requires careful attention to cellular context (existing cellular hardware).

In the design, fabrication, integration, and testing of new cellular hardware, synthetic biologists must use tools and methods derived from experimental biology. However, experimental biology still has not progressed to the point where it can provide an unshakable foundation for synthetic biology the way solid-state physics did for electrical engineering. As a result, design of synthetic biological systems has become an iterative process of modeling, construction, and experimental testing that continues until a system achieves the desired behavior. The process begins with the abstract design of devices, modules, or organisms, and is often guided by mathematical models. The synthetic biologist then tests the newly constructed systems experimentally. However, such initial attempts rarely yield fully functional implementations because of incomplete biological information. Rational redesign based on mathematical models improves system behavior in such situations. Directed evolution is a complimentary approach, which can yield novel and unexpected beneficial changes to the system. These retooled systems are once again tested experimentally and the process is repeated as needed. Many synthetic biological systems have been engineered successfully in this fashion because the methodology is highly tolerant to uncertainty. Synthetic biology will benefit from further such development and the creation of new methods that manage uncertainty and complexity.

The engineering strategies of standardization, decoupling, and abstraction can also be useful tools for dealing with the complexity of living systems (Weiss *et al*, 1999; Weiss, 2001; Endy, 2005; Keasling, 2005). Standardization involves establishing definitions of biological functions and methods for identifying biological parts, as with the registry of standard biological parts (Knight, 2002). Efforts in systems biology targeted at classification and categorization of genome elements can help define biological functions. Decoupling involves the decomposition of complicated problems into simpler problems, implemented by breaking down complex systems into its simpler constituents, and separating design from fabrication. Abstraction includes establishing hierarchies of devices and modules that allow separation and limited exchange of information between levels, and developing redesigned and simplified devices and modules, as well as libraries of parts with compatible interfaces.

The above engineering strategies come from disciplines where components are well behaved, easy to isolate from each other, and can subsist in isolation. The strategies must be adapted to work well in the biological realm, where biological components cannot exist without being connected. It may not be possible or desirable to fully separate and insulate biological devices and modules from each other and from the machinery of the host cells. The notions of standardization, decoupling, and abstraction must therefore be recast to better reflect the complexity of the cellular context. We will require not only standards for specific biological functions, but also standards for the states of contextual cellular elements

(e.g. ATP used when a cell divides). Although decoupling design from fabrication is useful, breaking down complex systems into many simpler ones will miss the connections between the simple elements and the radical interconnectedness of cellular context with each inserted module. Accordingly, our abstractions of device and module function must include cellular context.

A biological device has no meaning isolated from a module; a module has no meaning isolated from a cell; a cell has no meaning isolated from a population of cells. This contextual dependence is an essential feature of living systems and is not an impasse, but rather a bridge to the successful engineering of living systems. As with the uncertainty principle in quantum mechanics, it may be prudent to treat some biological uncertainties as fundamental properties of individual cell behavior (e.g. gene expression noise, context dependence, fluctuating environments). The fact that we always use populations of synthetic cells to complete tasks means that the criteria of reliability and predictability should apply at the cell population level. As long as a significant number of the cell population performs our desired task, the unpredictability of events occurring at the molecular level should have minimal effect. Design and fabrication methods that take into account uncertainty and context dependence will likely lead to on-demand, just-in-time customization of biological devices and components, which need not behave perfectly. Building imperfect systems is acceptable, as long as they perform tasks adequately. Synthetic biology should use the strategies that make biological systems versatile and robust as part of its own design practices. The success of synthetic biology will depend on its capacity to surpass traditional engineering, blending the best features of natural systems with artificial designs that are extensible, comprehensible, user-friendly, and most importantly implement stated specifications to fulfill user goals.

References

- Alon U (2003) Biological networks: the tinkerer as an engineer. *Science* **301**: 1866–1867
- Anderson JC, Clarke EJ, Arkin AP, Voigt CA (2005) Environmentally controlled invasion of cancer cells by engineered bacteria. *J Mol Biol* **355**: 619–627
- Atkinson MR, Savageau MA, Myers JT, Ninfa AJ (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell* **113**: 597–607
- Balagadde FK, You L, Hansen CL, Arnold FH, Quake SR (2005) Long-term monitoring of bacteria undergoing programmed population control in a microchemostat. *Science* **309**: 137–140
- Baron U, Gossen M, Bujard H (1997) Tetracycline-controlled transcription in eukaryotes: novel transactivators with graded transactivation potential. *Nucleic Acids Res* **25**: 2723–2729
- Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular system for programmed pattern formation. *Nature* **434**: 1130–1134
- Basu S, Mehreja R, Thiberge S, Chen MT, Weiss R (2004) Spatiotemporal control of gene expression with pulse-generating networks. *Proc Natl Acad Sci USA* **101**: 6355–6360
- Bayer TS, Smolke CD (2005) Programmable ligand-controlled riboregulators of eukaryotic gene expression. *Nat Biotechnol* **23**: 337–343
- Becskei A, Serrano L (2000) Engineering stability in gene networks by autoregulation. *Nature* **405**: 590–593
- Blake WJ, Kaern M, Cantor CR, Collins JJ (2003) Noise in eukaryotic gene expression. *Nature* **422**: 633–637
- Braun D, Basu S, Weiss R (2005) Parameter estimation for two synthetic gene networks: a case study. *IEEE International Conference on Acoustics, Speech, and Signal Processing* March 2005
- Bulter T, Lee SG, Wong WW, Fung E, Connor MR, Liao JC (2004) Design of artificial cell-cell communication using gene and metabolic networks. *Proc Natl Acad Sci USA* **101**: 2299–2304
- Carr PA, Park JS, Lee YJ, Yu T, Zhang S, Jacobson JM (2004) Protein-mediated error correction for *de novo* DNA synthesis. *Nucleic Acids Res* **32**: e162
- Cello J, Paul AV, Wimmer E (2002) Chemical synthesis of poliovirus cDNA: generation of infectious virus in the absence of natural template. *Science* **297**: 1016–1018
- Chan LY, Kosuri S, Endy D (2005) Refactoring bacteriophage T7. *Mol Syst Biol* doi:10.1038/msb4100025
- Chen MT, Weiss R (2005) Artificial cell-cell communication in yeast *Saccharomyces cerevisiae* using signaling elements from *Arabidopsis thaliana*. *Nat Biotechnol* **23**: 1551–1555
- Collins CH, Arnold FH, Leadbetter JR (2005) Directed evolution of *Vibrio fischeri* LuxR for increased sensitivity to a broad spectrum of acyl-homoserine lactones. *Mol Microbiol* **55**: 712–723
- Dueber JE, Yeh BJ, Chak K, Lim WA (2003) Reprogramming control of an allosteric signaling switch through modular recombination. *Science* **301**: 1904–1908
- Dwyer MA, Hellinga HW (2004) Periplasmic binding proteins: a versatile superfamily for protein engineering. *Curr Opin Struct Biol* **14**: 495–504
- Dwyer MA, Looger LL, Hellinga HW (2004) Computational design of a biologically active enzyme. *Science* **304**: 1967–1971
- Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* **403**: 335–338
- Elowitz MB, Levine AJ, Siggia ED, Swain PS (2002) Stochastic gene expression in a single cell. *Science* **297**: 1183–1186
- Endy D (2005) Foundations for engineering biology. *Nature* **438**: 449–453
- Farmer WR, Liao JC (2000) Improving lycopene production in *Escherichia coli* by engineering metabolic control. *Nat Biotechnol* **18**: 533–537
- Feng XJ, Hooshangi S, Chen D, Li G, Weiss R, Rabitz H (2004) Optimizing genetic circuits by global sensitivity analysis. *Biophys J* **87**: 2195–2202
- Fung E, Wong WW, Suen JK, Bulter T, Lee SG, Liao JC (2005) A synthetic gene-metabolic oscillator. *Nature* **435**: 118–122
- Garcia-Ojalvo J, Elowitz MB, Strogatz SH (2004) Modeling a synthetic multicellular clock: repressilators coupled by quorum sensing. *Proc Natl Acad Sci USA* **101**: 10955–10960
- Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in *Escherichia coli*. *Nature* **403**: 339–342
- Giesecke AV, Fang R, Joung JK (2006) Synthetic protein-protein interaction domains created by shuffling Cys₂His₂ zinc fingers. *Mol Syst Biol* doi:10.1038/msb4100053
- Guntas G, Mansell TJ, Kim JR, Ostermeier M (2005) Directed evolution of protein switches and their application to the creation of ligand-binding proteins. *Proc Natl Acad Sci USA* **102**: 11224–11229
- Guntas G, Ostermeier M (2004) Creation of an allosteric enzyme by domain insertion. *J Mol Biol* **336**: 263–273
- Hooshangi S, Thiberge S, Weiss R (2005) Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *Proc Natl Acad Sci USA* **102**: 3581–3586
- Howard PL, Chia MC, Del Rizzo S, Liu FF, Pawson T (2003) Redirecting tyrosine kinase signaling to an apoptotic caspase pathway through chimeric adaptor proteins. *Proc Natl Acad Sci USA* **100**: 11267–11272
- Isaacs FJ, Dwyer DJ, Ding C, Pervouchine DD, Cantor CR, Collins JJ (2004) Engineered riboregulators enable post-transcriptional control of gene expression. *Nat Biotechnol* **22**: 841–847
- Kaern M, Blake WJ, Collins JJ (2003) The engineering of gene regulatory networks. *Annu Rev Biomed Eng* **5**: 179–206

- Karig D, Weiss R (2005) Signal-amplifying genetic circuit enables in vivo observation of weak promoter activation in the Rhl quorum sensing system. *Biotechnol Bioeng* **89**: 709–718
- Keasling J (2005) The promise of synthetic biology. *Bridge Natl Acad Eng* **35**: 18–21
- Knight Jr TF (2002) DARPA BioComp Plasmid Distribution 1.00 of Standard Biobrick Components. *Synthetic Biology Working Group Reports*, MIT
- Kobayashi H, Kaern M, Araki M, Chung K, Gardner TS, Cantor CR, Collins JJ (2004) Programmable cells: interfacing natural and engineered gene networks. *Proc Natl Acad Sci USA* **101**: 8414–8419
- Kodumal SJ, Patel KG, Reid R, Menzella HG, Welch M, Santi DV (2004) Total synthesis of long DNA sequences: synthesis of a contiguous 32-kb polyketide synthase gene cluster. *Proc Natl Acad Sci USA* **101**: 15573–15578
- Kodumal SJ, Santi DV (2004) DNA ligation by selection. *Biotechniques* **37**, 34, 36, 38 passim
- Kramer BP, Fischer C, Fussenegger M (2004) BioLogic gates enable logical transcription control in mammalian cells. *Biotechnol Bioeng* **87**: 478–484
- Looger LL, Dwyer MA, Smith JJ, Hellinga HW (2003) Computational design of receptor and sensor proteins with novel functions. *Nature* **423**: 185–190
- Lutz R, Bujard H (1997) Independent and tight regulation of transcriptional units in *Escherichia coli* via the LacR/O, the TetR/O and AraC/11-12 regulatory elements. *Nucleic Acids Res* **25**: 1203–1210
- Lutz R, Lozinski T, Ellinger T, Bujard H (2001) Dissecting the functional program of *Escherichia coli* promoters: the combined mode of action of Lac repressor and AraC activator. *Nucleic Acids Res* **29**: 3873–3881
- Mangan S, Alon U (2003) Structure and function of the feed-forward loop network motif. *Proc Natl Acad Sci USA* **100**: 11980–11985
- Mangan S, Zaslaver A, Alon U (2003) The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. *J Mol Biol* **334**: 197–204
- Martin VJ, Pitera DJ, Withers ST, Newman JD, Keasling JD (2003) Engineering a mevalonate pathway in *Escherichia coli* for production of terpenoids. *Nat Biotechnol* **21**: 796–802
- McAdams HH, Arkin A (1998) Simulation of prokaryotic genetic circuits. *Annu Rev Biophys Biomol Struct* **27**: 199–224
- McMillen D, Kopell N, Hasty J, Collins JJ (2002) Synchronizing genetic relaxation oscillators by intercell signaling. *Proc Natl Acad Sci USA* **99**: 679–684
- Miller MB, Bassler BL (2001) Quorum sensing in bacteria. *Annu Rev Microbiol* **55**: 165–199
- Milo R, Itzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M, Alon U (2004) Superfamilies of evolved and designed networks. *Science* **303**: 1538–1542
- Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. *Science* **298**: 824–827
- Monod J, Jacob F (1961) Teleonomic mechanisms in cellular metabolism, growth, and differentiation. *Cold Spring Harb Symp Quant Biol* **26**: 389–401
- Park SH, Zarrinpar A, Lim WA (2003) Rewiring MAP kinase pathways using alternative scaffold assembly mechanisms. *Science* **299**: 1061–1064
- Pennisi E (2005) Synthetic biology. Synthetic biology remakes small genomes. *Science* **310**: 769–770
- Ptashne M (1986) *A Genetic Switch: Phage Lambda and Higher Organisms*. Cambridge, MA: Cell Press and Blackwell Scientific Publications
- Ronen M, Rosenberg R, Shraiman BI, Alon U (2002) Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proc Natl Acad Sci USA* **99**: 10555–10560
- Rosenfeld N, Alon U (2003) Response delays and the structure of transcription networks. *J Mol Biol* **329**: 645–654
- Rosenfeld N, Young JW, Alon U, Swain PS, Elowitz MB (2005) Gene regulation at the single-cell level. *Science* **307**: 1962–1965
- Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat Genet* **31**: 64–68
- Smith HO, Hutchison III CA, Pfannkoch C, Venter JC (2003) Generating a synthetic genome by whole genome assembly: phiX174 bacteriophage from synthetic oligonucleotides. *Proc Natl Acad Sci USA* **100**: 15440–15445
- Stemmer WP, Cramer A, Ha KD, Brennan TM, Heyneker HL (1995) Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxynucleotides. *Gene* **164**: 49–53
- Thattai M, van Oudenaarden A (2001) Intrinsic noise in gene regulatory networks. *Proc Natl Acad Sci USA* **98**: 8614–8619
- Tian J, Gong H, Sheng N, Zhou X, Gulari E, Gao X, Church G (2004) Accurate multiplex gene synthesis from programmable DNA microchips. *Nature* **432**: 1050–1054
- Weinberger LS, Burnett JC, Toettcher JE, Arkin AP, Schaffer DV (2005) Stochastic gene expression in a lentiviral positive-feedback loop: HIV-1 Tat fluctuations drive phenotypic diversity. *Cell* **122**: 169–182
- Weiss R (2001) *Cellular Computation and Communications using Engineered Genetic Regulatory Networks*. Cambridge, MA: Massachusetts Institute of Technology
- Weiss R, Basu S (2002) The device physics of cellular logic gates. *NSC-1: The First Workshop of Non-Silicon Computing*, Boston, MA
- Weiss R, Homsy G, Knight TF (1999) Toward *in-vivo* digital circuits. *Dimacs Workshop on Evolution as Computation*, Princeton, NJ
- Weiss R, Knight TF (2000) Engineered communications for microbial robotics. *DNA6: Sixth International Meeting on DNA Based Computers*, pp 1–55
- Yokobayashi Y, Weiss R, Arnold FH (2002) Directed evolution of a genetic circuit. *Proc Natl Acad Sci USA* **99**: 16587–16591
- You L, Cox III RS, Weiss R, Arnold FH (2004) Programmed population control by cell-cell communication and regulated killing. *Nature* **428**: 868–871
- Zimmer C (2003) Genomics. Tinker, tailor: can Venter stitch together a genome from scratch? *Science* **299**: 1006–1007